Defense Technical Information Center Compilation Part Notice

ADP014184

TITLE: Nonlinear Computational Aeroelasticity: Formulations and Solution Algorithms

DISTRIBUTION: Approved for public release, distribution unlimited
Availability: Hard copy only.

This paper is part of the following report:

TITLE: Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation [La reduction des couts et des delais d'acquisition des vehicules militaires par la modelisation avancee et la simulation de produit virtuel]

To order the complete compilation report, use: ADA415759

The following component part numbers comprise the compilation report:
ADP014142 thru ADP014198

Nonlinear Computational Aeroelasticity: Formulations and Solution Algorithms

A. Soulaïmani, A. BenElhajAli and Z. Feng
Département de génie mécanique
École de technologie supérieure
1100 Notre-Dame Ouest
Montréal, Québec
Canada, H3C 1K3

E-mail : asoulaimani@mec.etsmtl.ca

Abstract

This paper highlights some technical features of an analysis methodology being developed for nonlinear computational aeroelasticity. A conservative finite element formulation for the coupled fluid/mesh interaction problem is proposed. Fluid-structure coupling algorithms are then discussed with some emphasis on distributed computing strategies. Numerical results are finally shown for the Agard 445.6 wing.

Introduction

Aeroelasticity is the study of the mutual interactions between aerodynamic, inertial and elastic forces on flexible structures, such as aircraft. The aerodynamic forces induced by the flow on an aircraft depend on the geometric configuration of the structure. On the other hand, the aerodynamic forces cause elastic deformations and displacements of the structure. Accurate prediction of aeroelastic phenomena such as static divergence and flutter is essential to the *design and control* of high performing and *safe* aircraft. Transport aircraft of the future are expected to become much more complex. With the advanced subsonic and the transonic civil aircraft, it is becoming increasingly important to perform static and dynamic aeroelastic analysis using highly accurate fluid and structural computational models. In general, classical aeroelasticity often leads to oversized aircraft design. Thus, more accurate computational capabilities for aeroelascitity analysis are desired. In general, aeroelasticity analysis treats static and dynamic aspects. Static analysis is usually associated with performance. In dynamic analysis, the concern is focused on safety through *stability*, and *dynamic response* studies. Instability problems occur when the structure sustains energy from the fluid that exceeds the capacity of elastic potential energy. Thus, there exists a critical flight speed beyond which instabilities take place that are characterized by high amplitude oscillations. Wing *flutter* is an example of such instabilities. *Buffeting*, which is the unsteady response of a structure caused by the fluctuations in the incoming flow, is another example. In aeroelastic response problems, one looks for the deformation and stress states in the structure as a response to turbulence or any unsteadiness in the flow. When the response of the structure is finite, the structure is stable. The structure flutters when its response to any finite disturbance is highly amplified.

We present an integrated CFD-CSD simulation methodology for flutter calculations based on distributed-parallel finite elements solvers. The main technical features of the proposed approach are:

- **Flow solver:** Stabilized Finite Element Formulations are being used for space discretization of the three-dimensional Euler and Navier-Stokes equations. An implicit parallel solver based on Schwarz Domain Decomposition methods and on the nonlinear version of the GMRES algorithm is developed.

Paper presented at the RTO AVT Symposium on "Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation", held in Paris, France, 22-25 April 2002, and published in RTO-MP-089.

- **Mesh solver:** Arbitrary-Lagrangian-Eulerian kinematics formulation is used in order to adapt the grid motion with the structure displacement. An ALE formulation is developed where, at any instant, the mesh configuration verifies the discrete form of the Geometric Conservation Law (DGCL). Furthermore, the dynamic mesh is modeled as a linear elastic material which undergoes large displacements.
- **Structure solver:** Finite Element commercial models for linear elasticity are used to perform the eigenvalue analysis. The time dependent structural displacement field is then computed using a classical Newmark scheme.
- **Distributed/Parallel computations:** Aeroelastic analyses are computationally intensive. Therefore, they can benefit from parallel processing technology. Intra parallelism (i.e., within each field) and inter parallelism (i.e., to couple the three fields fluid-mesh-structure) is performed using the message-passing paradigm. The computer code developed is designed to run on shared memory machines as well as on distributed machines such as Beowulf clusters. A Beowulf cluster recently developed at ETS containing 24 PCs and 12 GB of RAM is used presently as the distributed computing platform.
- **Fluid-Structure coupling:** Implicit solution algorithms are proposed to solve the CFD-CSD-Mesh coupled problem. It is based on the use of the Newton-GMRES algorithm for the entire problem along with a block preconditioning technique. Each block corresponds to *a functional* domain.
- **Matcher:** A fourth module is built in order to perform the tasks of load transfer from the fluid to the structure and the exchange of structure motion to the fluid.

The computational fluid dynamics code

PFES is our finite element code for the numerical solution of some multi-physics problems. For Euler and Navier-Stokes equations, a new formulation referred to as EBS (Edge Based Stabilized finite element formulation) is developed. This method combines in some way the best features of respectively the Galerkin finite element method (provides high order schemes, easy to implement on unstructured grids, etc.) and the Finite Volume based methods (e.g. ease to construct monotone upwind schemes on unstructured meshes). It was numerically demonstrated [1-3] that EBS can be less diffusive than SUPG [4-6] and the standard Finite Volume schemes. Accuracy is critical for solving shocks and separation regions present in the transonic regime. The implicit solver used is based on the Flexible GMRES algorithm preconditioned by the incomplete factorization ILUT [7]. The parallel version of the code makes use of the domain decomposition approach.

The Euler compressible equations in fixed meshes are written in a compact form and in terms of the conservation variables are written in a vector form as

$$\boldsymbol{V}_{,t}+\boldsymbol{F}_{i,i}^{adv}(\boldsymbol{V})=\boldsymbol{F}_{i,i}^{diff}(\boldsymbol{V})+\boldsymbol{F}_s \qquad (1)$$

where $\mathbf{V}$ is the vector which components are the specific momentum, density and specific total energy. F_i^{adv} and F_i^{diff} are respectively the convective and diffusive fluxes in the ith-space direction, and $\mathbf{F_s}$ is the source vector. Lower commas denote partial differentiation and repeated indices indicate summation. Diagonalizable Jacobian matrices can represent the convective fluxes $\boldsymbol{A}_i=\boldsymbol{F}_i^{adv},_{\boldsymbol{V}}$. Recall that any linear combination of these matrices has real eigenvalues and a complete set of eigenvectors. It is well known that the standard Galerkin finite element formulation often leads to numerical instabilities for convective dominated flows. In the Galerkin-Least-Squares method (or the generalized Streamline Upwind Petrove Galerkin method), the Galerkin variational formulation is modified to include and integral form depending on the local residual $\mathbf{R(V)}$ of equation (1), i.e. $\boldsymbol{R}(\boldsymbol{V})=\boldsymbol{V}_{,t}+\boldsymbol{F}_{i,i}^{adv}(\boldsymbol{V})-\boldsymbol{F}_{i,i}^{diff}(\boldsymbol{V})-\boldsymbol{F}_s$, which is identical to zero for the exact solution. The SUPG formulation reads: find $\mathbf{V}$ such that for all weighting functions $\mathbf{W}$,

$$\sum_e \int_{\Omega_e}\left[W.(V_{,t}+F_{i,i}^{adv}-F_s)+W_{,i}F_i^{diff}\right]d\Omega \qquad (2)$$

$$-\int_\Gamma W\cdot F_i^{diff}n_i\,d\Gamma-\int_\Gamma W.A_{\tilde{n}}(V-V_\infty)\,d\Gamma+\int_\Omega \mu_c\nabla\mathrm{W}.\nabla\mathrm{V}d\Omega+\sum_e\int_{\Omega e}\mathrm{A}_i^{\mathrm{t}}.\mathrm{W}_{,i}\;\tau\cdot\mathrm{R}(V)\;d\Omega=0$$

In this formulation, the matrix τ is referred to as *the matrix of time scales*. The SUPG formulation is built as a combination of the standard Galerkin integral form and a perturbation-like integral form depending on the

local residual vector. The third integral term in (2) takes into account the far field boundary conditions and the forth integral is a stabilizing term for the shocks. Recently, a new method referred to as the Edge-Based-Stabilized finite element method (EBS) was introduced to stabilize the standard Galerkin method while considering the real characteristics of the flow as computed on the normal direction of element edges. This method has been proven to be stable and accurate for solving viscous and inviscid compressible flows, but more time consuming as compared to SUPG. Let us now briefly present the EBS formulation.

Consider the eigen-decomposition of $\boldsymbol{A}_n=\sum \boldsymbol{A}_i \boldsymbol{n}_i, \boldsymbol{A}_n=\boldsymbol{S}_n \boldsymbol{\Lambda}_n \boldsymbol{S}_n^{-1}$.

Let $Pe_i=\lambda_i h/2\nu$ be the local Peclet number for the eigenvalue λ_i, h a measure of the element size on the element boundary, ν the physical viscosity and $\beta_i = \min (Pe_i/3, 1.0)$. We define the matrix $\boldsymbol{B}_n$ by

$$\boldsymbol{B}_n=\boldsymbol{S}_n \boldsymbol{L} \boldsymbol{S}_n^{-1} \tag{3}$$

where L is a diagonal matrix whose entries are given by $\boldsymbol{L}_i = (1 + \beta_i)$ if $\lambda_i > 0$; $\boldsymbol{L}_i = - (1 - \beta_i)$ if $\lambda_i < 0$ and $L_i = 0$ if $\lambda_i = 0$. The proposed EBS formulation is similar to (2) but the last integral term is replaced by:

$$+\sum_e \int_{\Gamma_e} \boldsymbol{W} \cdot \boldsymbol{\tau}_n^{ed} \cdot \boldsymbol{R}(\boldsymbol{V}) d\Gamma=0 \tag{4}$$

with τ_n^{ed} the matrix of *intrinsic length scales* given by $\boldsymbol{\tau}_n^{ed}=\frac{h}{2} \cdot \boldsymbol{B}_n$ (5)

Geometrically conservative ALE formulation

One of the considerations in the mathematical formulation of conservation laws is the type of the kinematic description used for the material particles. A Eulerian description is very often used in fluid dynamics, while a Lagrangian is common in solid mechanics. When the material body contains moving boundaries, a mixed description, partially Lagrangian and partially Eulerian (also called Arbitrary Lagrangian-Eulerian), is more convenient [8-9]. This occurs for any flexible structure surrounded by a flow. It therefore becomes necessary to solve the fluid equations on a moving grid in order to match the fluid and the structure boundaries. Thus, besides the fluid and the structure material fields, there is a third field constituted by the moving mesh, which can be viewed as a material body having its own motion and dynamics. The three-field coupled problem is constituted by a set of partial differential equations, which are coupled through boundary conditions. A class of solution procedures called partitioned or segregated has been advocated to solve this coupled problem [10-14]. Given the displacements of the nodes on the wing, i.e. after solving the structure displacement field, a differential elliptic operator is designed to distribute the boundary motion inside the domain in order to avoid nodes collapsing or elements degenerating. After updating the mesh configuration, the flow field is solved on this mesh. It is shown [10] that the algorithm constructed for updating the dynamic mesh must obey a discrete Geometric Conservation Law (DGCL). A physical interpretation of the GCL is that the motion of the nodes must be compatible with the fact that the volume swept by the edges of the control volume or the element should be exactly equal to the variation of its volume (i.e. volume preserving). Our interpretation of the GCL goes as follows. The DGCL is equivalent to satisfying the kinematic Euler equation for the mesh, i.e.

$$\frac{\partial \boldsymbol{J}}{\partial t}=\boldsymbol{J} \, div \, \boldsymbol{w} \tag{6}$$

where $\boldsymbol{J}$ is the determinant of the geometric gradient tensor $\boldsymbol{F}$ from the current mesh configuration to the reference one and **w** is the velocity of a point of the moving domain. Equation (6) holds for the continuous medium. However, when applying discretization methods in space (FE, FD or FVM) and in time, it is not a priori satisfied due to time and space discretization errors. Those errors should not spoil the accuracy of the coupled problem. In [10,12,13], special time integration schemes satisfying the DGCL for first and second order time accuracy have been proposed in the context of FV methods. Substantial modifications to the original code, which has been developed for fixed meshes, are then required. Given the above interpretation, we propose to investigate the DGCL in another way. The mesh is not updated unless it satisfies equation (6) in a discrete form. Then, a finite element formulation can be easily constructed to solve the operator for distributing the boundary displacement inside the domain constrained to satisfying, in a weak form, the Piola-Kirchoff equation. By doing this, standard time integration procedures usually used for fixed meshes are

accurate and still valid for dynamic meshes. Thus, CFD codes built for rigid meshes could be still valid for dynamic meshes. This idea is detailed in the following.

The conservation equations (1) in moving meshes are rewritten as

$$(JV)_{,t}+J\left(F_i^{adv}-w_iV\right)_{,i}=J\left(F_{i,i}^{diff}+F_s\right) \tag{7}$$

In the above equation, space differentiations are done with respect to the actual coordinates at the current time. Using simple differentiation operations, (7) is transformed into

$$(J_{,t}-J\,div\,w)V+J\,(V_{,t}+F_{i,i}^{adv}-w_iV_{,i}-F_{i,i}^{diff}-F_s)=0 \tag{8}$$

and the corresponding classical Galerkin variational form is :

$$\int_{\Omega_0}W\cdot\left(J_{,t}-J\,div\,w\right)V\,d\Omega+\int_{\Omega}W\cdot\left(V_{,t}-F_{i,i}^{adv}-w_i\cdot V_{,i}-F_s\right)d\Omega+\int_{\Omega}W_{,i}F_i^{diff}\,d\Omega-\int_{\Gamma}W\cdot F_i^{diff}n_i\,d\Gamma=0 \tag{9}$$

where Ω_0 is the configuration of the mesh at a reference time t_0 and Ω is the current configuration of the mesh at time t. For the continuous solution, Euler identity (6) is satisfied at any instant and for every point of the domain so that the first integral in (9) is identically zero. Thus, the variational formulation of the conversation equations is similar to the case of fixed mesh (up to the additional convective term $-w_iV_{,i}$ or in other words the advective matrices A_i are replaced in the case of moving domains by A_i-w_iI with I the identity matrix):

$$\int_{\Omega}W\cdot\left(V_{,t}-F_{i,i}^{adv}-w_i\cdot V_{,i}-F_s\right)d\Omega+\int_{\Omega}W_{,i}F_i^{diff}\,d\Omega-\int_{\Gamma}W\cdot F_i^{diff}n_i\,d\Gamma=0 \tag{10}$$

For a discrete numerical solution Euler identity is not necessarily satisfied. Thus the first integral term in (9) could not, in principle, be neglected. The usual DGCL condition states that for a dynamic mesh and for any arbitrary constant flow field Vc and for the case of $\mathbf{F_s}$ = 0, the solution of the discrete problem should be exactly Vc , thus (9) gives

$$\int_{\Omega0}\phi\left(J_{,t}-J\,div\,w\right)d\Omega=0 \tag{11}$$

where ϕ is any weighting scalar function. Equation (11) is simply the variational form corresponding to the constraint (6). Note that (10) actually satisfies the usual DGCL condition discussed above. More generally, if it is desired for any purpose to use the non-conservative variational formulation (10) while satisfying explicitly the DGCL condition, the mesh motion could be subjected to the constraint (11). In the finite element methodology one usually interpolates **w** by continuous piece-wise functions, so that the mesh velocity field is continuous in the continuous medium (i.e. the mesh). Then, equation (6) is satisfied for the exact time differentiation of J. However, applying a time discretization scheme, similar to that used for the fluid, to the mesh coordinates to obtain **w** and to $\frac{\partial J}{\partial t}$ yields a truncation error $\frac{\partial J}{\partial t}-J\,div\,w=o(\Delta t^p)$ which is consistent with the truncation error obtained for the discrete (fluid) conservation equations. Thus, one can adopt formulation (10) along with an appropriate time discretization scheme for the evaluation of **w** , and the stability and convergence in time are expected to be obtained. On the other hand, recent theoretical studies by Letallec and his group [15] showed the impact of DGCL on the conservation of energy of the coupled fluid-structure system considered as a unique continuous medium. Energy conservation is a key point in studying fluid-structure interactions. In particular, the evolution of the kinetic energy must be controlled. Most time integration schemes do violate this principle of energy conservation when dealing with deformable domains. More precisely, for fully coupled schemes using conservative formulations and non-volume preserving grid configuration (DGCL), a small pollution term appears in the kinetic energy principle, which may grow exponentially in time. More specifically, it can be shown that using the first order Euler time discretization, the scheme is volume preserving if the fluid equations are integrated over a configuration Ω which is located at the mid distance between two successive configurations of the mesh [10] .

Mesh motion

Many choices can be considered in designing the operator that distributes the fluid-interface motion inside the moving domain. We consider that the mesh motion is defined by the elasto-static equations defined in the mesh configuration at time t :

$$\rho_m \mathbf{x}_{,tt} - div(\mathbf{P}(\mathbf{x})) = b \tag{12}$$

where ρ_m, **P** and **b** are fictitious density, the PK1 stress tensor and the body force. Equation (12) is solved for the mesh displacements **x** with the kinematics condition at the moving boundary $\mathbf{w} \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n}$ for inviscid flows, with **u** the fluid velocity. The mesh velocity **w** is computed using a finite difference scheme (similar to that used for the fluid) to the differential equation $\boldsymbol{w} = \boldsymbol{x}_{,t}$. We usually take ρ_m and b equal to zero.

As the mesh moves, it is not always guaranteed that all elements keep an acceptable shape for accurate CFD computations. Especially, small elements are prompt to large distortions. In order to preclude negative volumes or large distortions for small elements, a suitable constitutive law for the mesh medium should be designed.

We consider the mesh as an elastic material undergoing small strains and large rotations. Thus, the PK2 stress tensor $\boldsymbol{S} = \boldsymbol{P}.\ \boldsymbol{F}^{-t}$, (with **F** the deformation tensor) is linear with the Green strain tensor $\boldsymbol{E} = (\boldsymbol{F}^{t}\boldsymbol{F} - \boldsymbol{I})/2$, thus:

$$\boldsymbol{S} = \boldsymbol{C}\,\boldsymbol{E} \tag{13}$$

where $\boldsymbol{C}$ is the fictitious elastic modulo tensor which entries are of order $0(h^{-3})$ with h the element size.

In summary, equations (12) are solved for the finite displacements **x** between the mesh configurations at time t and time $t+\Delta t$ along with the constitutive relation (13) and the boundary conditions. The non-zero boundary conditions for the mesh equations are actually the imposed displacements at the moving boundary which are computed by the CSD module. Note that the mesh motion problem can generate a large system of nonlinear equations. These are solved, at every time step, using a preconditioned Krylov algorithm (CG or even GMRES).

The CSD analysis

The finite element method is well established for solid and structure computations. In industry, commercial codes are very often used for linear structure analysis. In this work, we consider a classical linear model for the structure which can be described by modal equations as

$$\ddot{z}_i(t) + 2\eta_i\,\omega\,\dot{z}_i(t) + \omega_i^2 z_i(t) = s_{0i}(t) \tag{14}$$

with z_i the generalized normal mode displacement, η_i and ω_i are respectively the damping and the natural frequency of the i^{th} mode. Newmark's algorithm is used to integrate (14).

Coupling algorithms and distributed computing

In dynamic response problems, one looks for the successive flow and structure behaviours for a given set of initial conditions, such as a perturbation in the flow. In linear theory, the flutter speed of an aircraft can be obtained directly from the solution of an eigenvalue problem. In the nonlinear theory, for a given set of flight conditions, predicting whether an aircraft will flutter or not is much more complex and computationally intensive. There are two possible approaches (a) Starting from a deformed state of the structure the fluid-structure coupled solution is computed in the time domain [10-13], (b) Starting around an initial equilibrium state an eigenvalue problem is established by linearizing the coupled dynamic system [14]. The first approach is simpler to implement and enables to capture all the nonlinearities in the fluid-structure system.

Implicit time marching schemes enable the use of large time steps for the structure as well as for the mesh and the fluid fields. The conventional partitioned procedure commonly used in fluid-structure interactions is illustrated in Figure 1. It is based on the following steps:

1. Update the fluid grid to conform to the structural boundary at time t_n .
2. Advance the flow field using the new boundary conditions.
3. Update the surface load on the structure based on the fluid solution at time t_{n+1}
4. Advance the structure using the new fluid surface load.

For parallel computing, one can use an inter-field partitioned approach as illustrated in Figure 2.

1. The fluid grid is updated to conform to the structural boundary at time t_n and the fluid is advanced using the structural boundary conditions at time t_n
2. The structure is advanced using the fluid surface load at time t_n.

With this procedure, the CFD and CSD solvers can run in parallel during the time interval $[t_n , t_{n+1}]$. Inter-field communication and I/O transfer is needed only at the beginning of each time interval. As time progress, there may be a lag between the fluid and the structure so that a spurious energy exchange at the interface may generates undesirable instabilities.

In the following, an implicit iterative scheme is proposed to enhance the coupling between the mesh, the structure and the flow fields. After time and space discretizations of the fluid, structure and mesh equations, an algebraic system of equations for the unknown variables $\mathbf{S= (V, x, q)^t}$ (i.e. flow quantities and structure-mesh displacements) is obtained, which can be formally written as

$$\boldsymbol{G(S)=0} \tag{15}$$

This non-linear system can be solved using Newton's method as follows:

1. Given an initial structure and mesh configurations and a flow field for the current time step t_n;
2. Do n=1, maximum number of time steps;
3. Do i= 1, maximum number of iterations;
4. Find the correction $\mathbf{\Delta S^i}$ solution of :

$$\boldsymbol{H\ \Delta S^i = -G\left(S^{i-1}\right)}$$

where **H** is the Jacobian matrix associated to **G**;

5. Check the convergence. If satisfied go to 6;
6. EndDo;
7. Update the global solution $\mathbf{S^i= S^{i-1}+\Delta S^i}$
8. EndDo.

In this algorithm, the Jacobian matrix **H** is needed. While it is difficult to develop its analytical expression, it is possible to compute an approximation. On the other hand, using Krylov-based iterative methods, such as the GMRES algorithm, to solve the linear system in step (4), one actually only need to compute the matrix-vector product of H and a direction vector z. A good approximation can be computed using a finite-difference formula like

$$\boldsymbol{Hz=\frac{G(V_0+\sigma z)-G(V_0)}{\sigma}} \tag{16}$$

where σ is a small scalar.

To be efficient, Krylov methods need to be appended by a preconditioned, which is in principle a good approximation of **H**. A straightforward choice would be to use a *block-diagonal* matrix, which entries are the approximate Jacobians associated with respectively the flow, the structure and the mesh fields. In other words, the coupled problem is seen as a set of non-linear equations obtained by discretizing a continuous medium.

These equations are solved iteratively using preconditioned Newton-GMRES method. While the global residual vector **G(S)** is needed, its main three components are computed by calling the corresponding modules (CFD solver, mesh solver or CSD solver). Obviously, these computations can be performed in parallel and inter-field communications are needed. This decomposition is referred to as the *functional decomposition*. On the other hand, the residuals of the flow field and the mesh motion are computed using a classical *domain decomposition approach*. For CFD, as well as for the mesh motion, we use a robust parallel iterative solver (i.e. intra-parallelism) based on Schwarz-Newton-Krylov techniques. Thus, the available processors are divided into groups of processors; each group is assigned to a specific field. Since CFD computations are more demanding, we assign more processors for the CFD domain than for the mesh solver.
Note that repeated updates of the geometry, for different flow conditions, are needed during the Newton-GMRES iterations. Thus, the flow field continuously drives the geometry so that a better conservation of the energy at the fluid-structure interface can be obtained. Since the main part of the computations is consumed within the CFD solver, the additional communications will not increase the overall simulation cost significantly, provided the number of time steps and non-linear CFD-iterations remains unchanged.
This algorithm is illustrated in Figure 3.

1. Given a geometry and a flow field at time t_n.
2. Compute the initial residual vector $\mathbf{R_0 = G(S_0)}$.
3. Perform Newton–GMRES iterations: Compute the residual for a perturbed solution in a direction **z**: $\mathbf{G(S_0 + \sigma z)}$; this has three major components each one is computed by its corresponding solver.
4. Perform inter-field communications.
5. Test for convergence and update the global solution for time step t_{n+1}.
6. Go to next step.
7.

Note that in the above algorithm one has to perform inter-filed communications in order to compute Krylov-directions for the global problem.
Several variants of this algorithm can be thought of. In its simplest form, one can perform a number of global iterations in which inter-filed communications occur only at the beginning of every iteration (i.e. this is nothing but the fixed-point algorithm). For a reasonable time step, one can observe the convergence for the fluid force transmitted to the structure and for the generalized coordinates after few global iterations.

Inter-field communications

Fluid-structure interactions involve the transfer of loads from the fluid mesh to the structure and the transfer of mesh structure motion to the moving mesh boundary. Since the CFD mesh is much finer than that used by the structure, the traces of these two meshes at the fluid-structure interface do not necessarily match. Then, load and displacements transfer cannot be done in a trivial way. The importance of conservative load transfer in fluid-structure interaction problems has recently been addressed in [16]. We adopt here the algorithm proposed in [16] which main steps consist of:

a) Pairing each fluid grid point S_j on the fluid interface Γ_f with the closet wet structural element $\Omega_S^{(e)} \in \Gamma_S$ (see Figure 4);
b) Determining the natural coordinates χ_j in $\Omega_S^{(e)}$ of the fluid point S_j (or its projection onto $\Omega_S^{(e)}$);
c) Interpolating the displacement of fluid nodes x_f inside $\Omega_S^{(e)}$ using the structure shape functions N_i^s ;
d) Projecting the generalized fluid force to the structure as:

$$f_i = \sum_{j=1}^{j=j_F} \Phi_j N_i^s(\chi_j)$$

with, $\Phi_j = \int_{\Gamma_f} (-pn + \sigma_f \cdot n) N_j^F d\gamma$

The generalized force associated to the fluid node S_j. It is proved that this algorithm preserves load and energy conservation at the fluid-structure interface. On the other hand, in aeroelasticity, the structure is often represented by plate, shell and beam elements which results in geometric discrepancies between the fluid mesh

skin Γ_F and the structure boundary Γ_S (as illustrated in Figure 4). Thus, the above transfer algorithm is preceded by a projection step of the fluid nodes **M** onto the structure elements to locate the point **P** and to compute the gap vector **PM**. The motion of P is found using the above interpolation procedure and the motion of M is obtained by considering that the vector **PM** rotates with the structural element as it is done in classical plate theory.

Numerical results

Numerical tests have been carried out for the popular AGARD-445.6 [17] wing. The AGARD-445.6 is a thin swept-back and tapered wing with a symmetrical NACA 65A004 airfoil section. The weakened model-3 is considered here. A coarse unstructured grid is employed for Euler computations that has 37965 nodes, 177042 elements and generates 388464 coupled equations. For the structure, a mesh of 1176 quadrilateral shell elements and 1250 nodes. Using the commercial software Ansys, the firs five dry modes have been computed. There respective frequencies are the following: 9.6 Hz, 39.42Hz, 49.60 Hz, 96.095Hz and 126.30 Hz. The shape modes compare well with those of Yates (figure 5). A flow at a free-stream Mach number of 0.96 and zero angle of attack is computed first. At t=0, a Dirac force is imposed on the tip of the wing. The response of the wet structure is computed for different free stream pressures q. We use a second order time differentiation scheme with a non dimensional time step $\Delta t = 0.2$ and three global iterations. The experimental flutter pressure at M= 0.96 is q = 61.3 lb/sq ft [17]. Figure 6 shows that at the computed conditions (M= 0.96 is q = 62.0 lb/sq ft) the structure is neutrally stable and the first two modes are in coalescence. Figure 7 shows a comparison of two models for the mesh motion (with q= 71.3), a nonlinear model as described previously (used the above computations) and a linear one where **F= I** and where the second order terms in **E** are dropped. In the linear model the mesh becomes distorted as the motion amplifies until it collapses at time step 290 (negative jacobians). With the nonlinear model, computations run for 900 time steps. Finally, figure 8 shows a comparison of the flutter boundary obtained with our code with the experimental observations [17] and with some numerical results reported in the literature[18 , 19 and 20].

Conclusion

In this paper we have presented a CFD-based aeroelastic model. A suitable finite element formulation is used for all computational fields (fluid, mesh and structure). A functional decomposition approach is used for the solution of the coupled problem. For every physical field a parallel GMRES algorithm is employed to solve the corresponding discrete system. Inter-filed communications occur during global quasi-Newton coupling iterations. Numerical tests on the Agard 445.6 aeroelastic wing show that the inter-filed communications strongly enhance the numerical stability of the time marching procedure. The flutter dip is well produced when three global coupling iterations are used. On the other hand, a nonlinear model for the moving mesh is proposed. Numerical tests show that this model improves the robustness of the aeroelastic code. Finally, an accurate flutter boundary is obtained for the Agard445.6 wing aeroelastic test case.

Acknowledgments

This research has been funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), PSIR research program of ETS and La Fondation J. Armand Bombardier.

References

[1] A. Soulaïmani and A. Rebaine. "An edge based stabilized finite element method for solving compressible flows", the 14th AIAA Computational Fluid Dynamics Conference, paper AIAA-99-3269.

[2] A. Soulaïmani and C. Farhat. "On a finite element method for solving compressible flows", ICES98, Atlanta, 1998.

[3] A. Soulaïmani, A. Rebaine and Y. Saad. "An edge based stabilized finite element method for solving compressible flows: formulation and parallel implementation", Comput. Methods Appl. Mech. Engrg, vol. 190, pp.6735-6761 (2001).

[4] T.J.R. Hughes and Mallet. "A new finite element formulation for computational fluid dynamics III. The generalized streamline operator for multidimensional advective-diffusive systems", Comput. Methods Appl. Mech. Engrg, 58, 305-328, 1986.

[5] A. Soulaïmani and M. Fortin. "Finite element solution of compressible viscous flows using conservative variables", Comput. Methods Appl. Mech. Engrg., 118, 319-350, 1994.

[6] N.E. Elkadri, A. Soulaïmani and C. Deschenes. "A finite element formulation of compressible flows using various sets of independent variables", Comput. Methods Appl. Mech. Engrg., 181, 161-189, 2000.

[7] Y. Saad. "Iterative Methods for Sparse Linear Systems", *Numerical Linear Algebra with Applications*, 1:387-402, 1994.

[8] J. Donea. "An arbitrary Lagrangian-Eulerian finite element method for transient fluid structure interactions", Comput. Meths. Appl. Mech. Engrg., 33, 689-723, 1982.

[9] A. Soulaïmani and Y. Saad: "An Arbitrary Lagrangian Eulerian finite element formulation for solving three-dimensional free surface flows", 162, 79-106, 1998.

[10] C. Farhat and M. Lesoinne. "On the accuracy, stability, and performance of the solution of three-dimensional nonlinear transient aeroelastic problems by partitioned procedures", AIAA-96-1388, 1996.

[11] R. Löhner et al. "Fluid-Structure Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids", AIAA Paper 95-2259.

[12] M. Lesoinne and C. Farhat. "Geometric conservation laws for aeroelastic computations using unstructured dynamic meshes", AIAA Paper 95-1709.

[13] C. Farhat and M. Lesoinne. "Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems", Comput. Maths. Appl. Mech. Engrg, 182, 499-515, 2000.

[14] M. Lesoinne and C. Farhat. "CFD-based aeroelastic eigensolver for the subsonic, transonic and supersonic regimes", Journal of Aircraft, vol 38, no 4, 2001.

[15] T. Fanion, M. Fernandez and P. LeTallec: "Deriving adequate formulations for fluid-structure interaction problems : from ALE to transpiration". Revue Européenne des éléments finis. Volume 9-6, pp. 681-70, 2000.

[16] C. Frahat, M. Lesoinne and P. LeTallec . "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity". Comput. Maths. Appl. Mech. Engrg, vol. 157, pp. 95-114, 1998.

[17] E.C. Yates. "AGARD Standard Aeroelastic Configuration for Dynamics Response, Candidat Configuration I.-Wing 445.6", NASA TM 100492, 1987.

[18] E.M. Lee-Rausch, J.T. Batina. "Wing-flutter boundary prediction using unsteady Euler equations aerodynamic methd". AIAA Paper No. 93-1422, 1993.

[19] K.K. Gupta. "Development of a finite elemnt aeroelastic analysis capability". J. Aircraft 33 (1996) 995-1002.

[20] M. Lesoinne, M. Sarkis, U. Hetmaniuk and C. Farhat. "A linearized method for the frequency analysis of three-dimensional fluid/structure interaction problems in all flow regimes". Comput. Maths. Appl. Mech. Engrg. 190 (2001) 3121-3146.

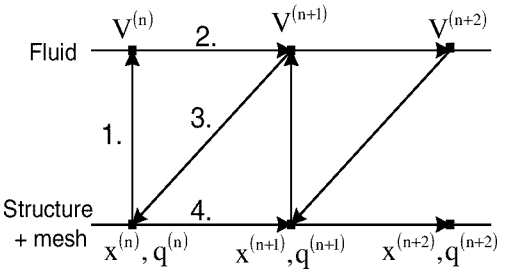

Figure 1. Conventional partitioned procedure

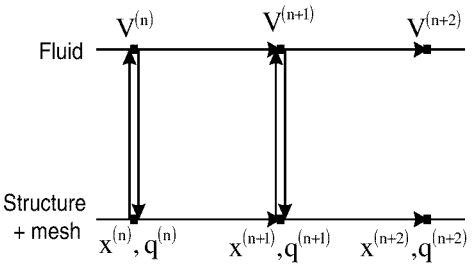

Figure 2. Inter-field partitioned procedure

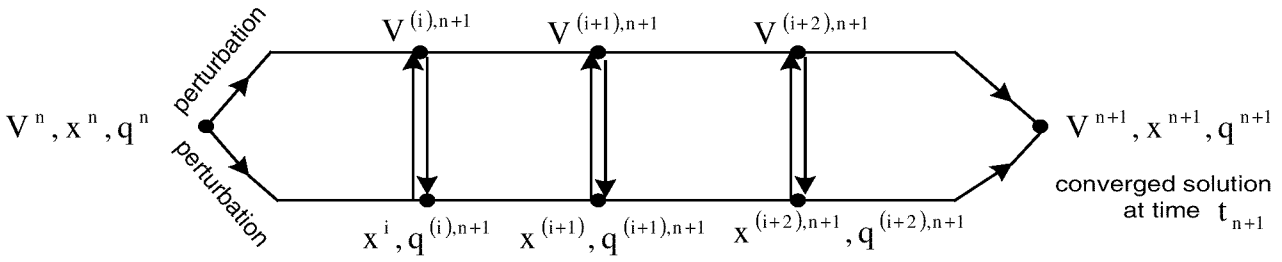

Figure 3. A fully implicit coupled procedure

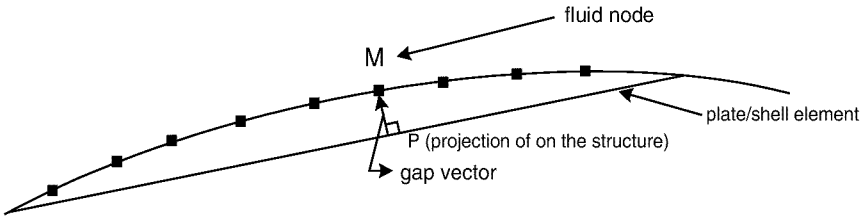

Figure 4. Fluid-solid interface moves according to plate theory

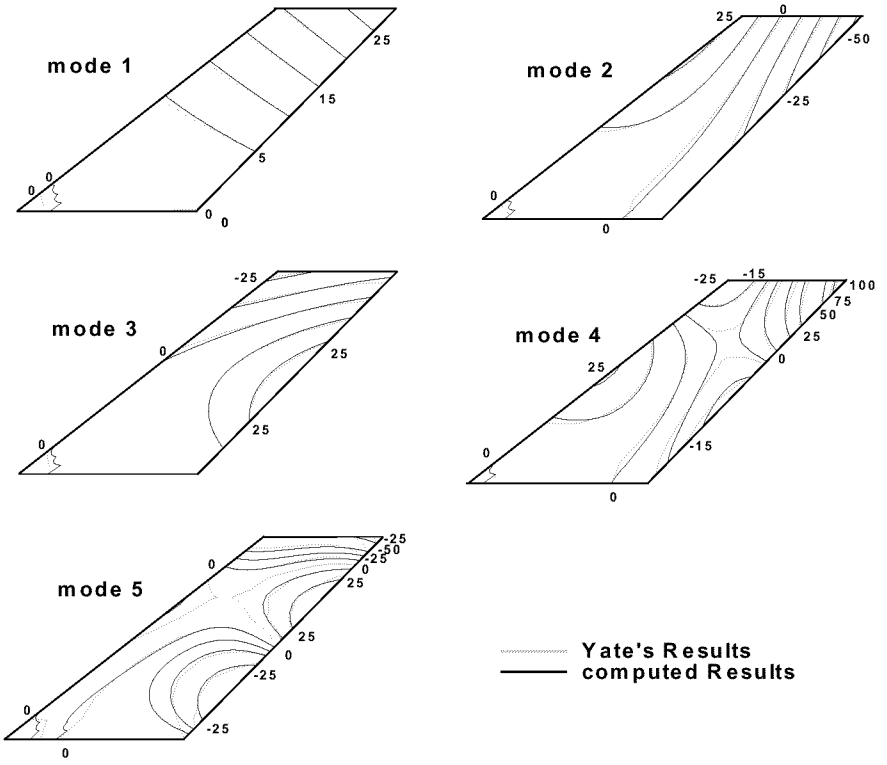

Figure 5. Agard 445.6 structure shape modes

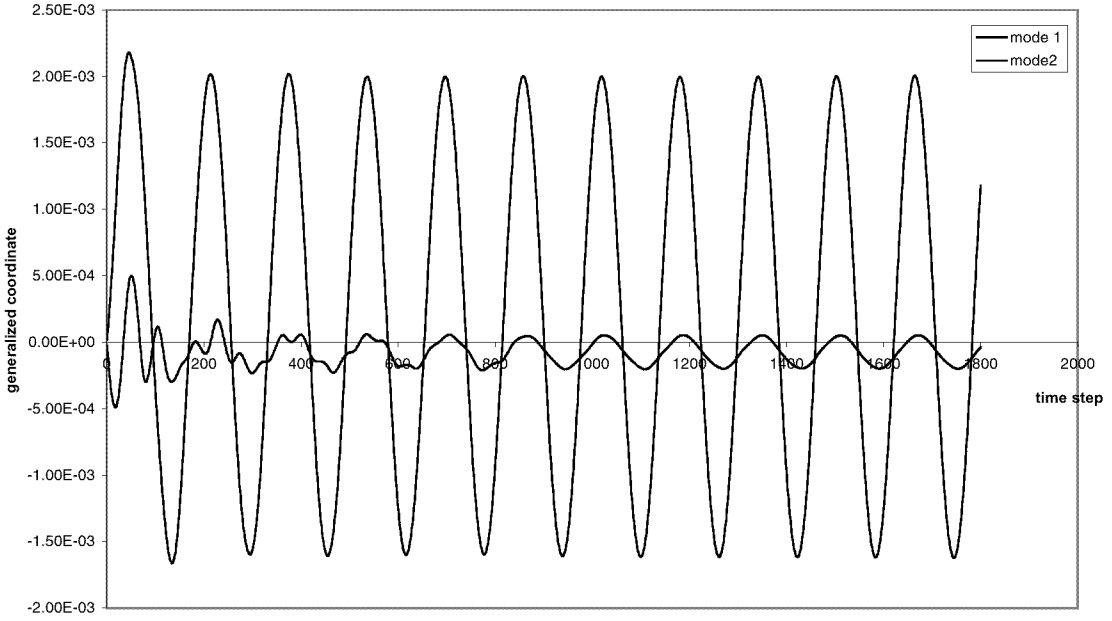

Figure 6. Time history for the first and second modes for Mach= 0.96 and q= 62 lb /sq ft and using three coupling iterations

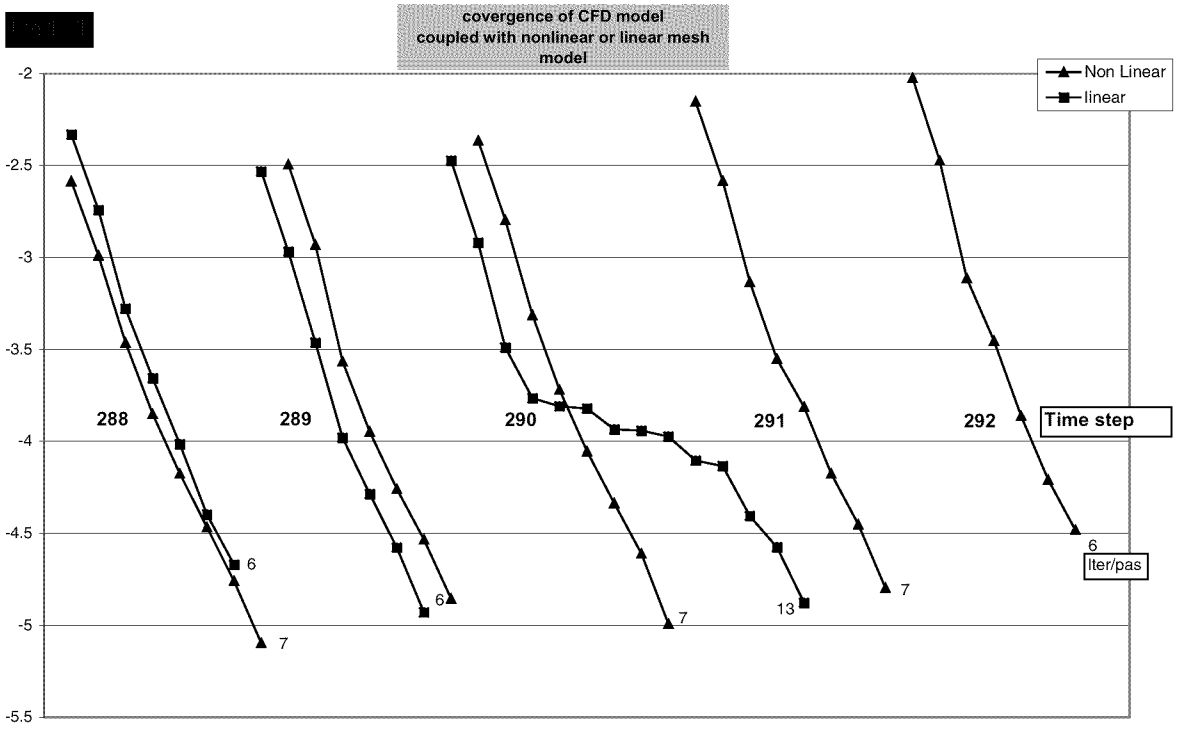

Figure 7. Convergence history for the CFD computations using respectively. A linear and a nonlinear mesh model

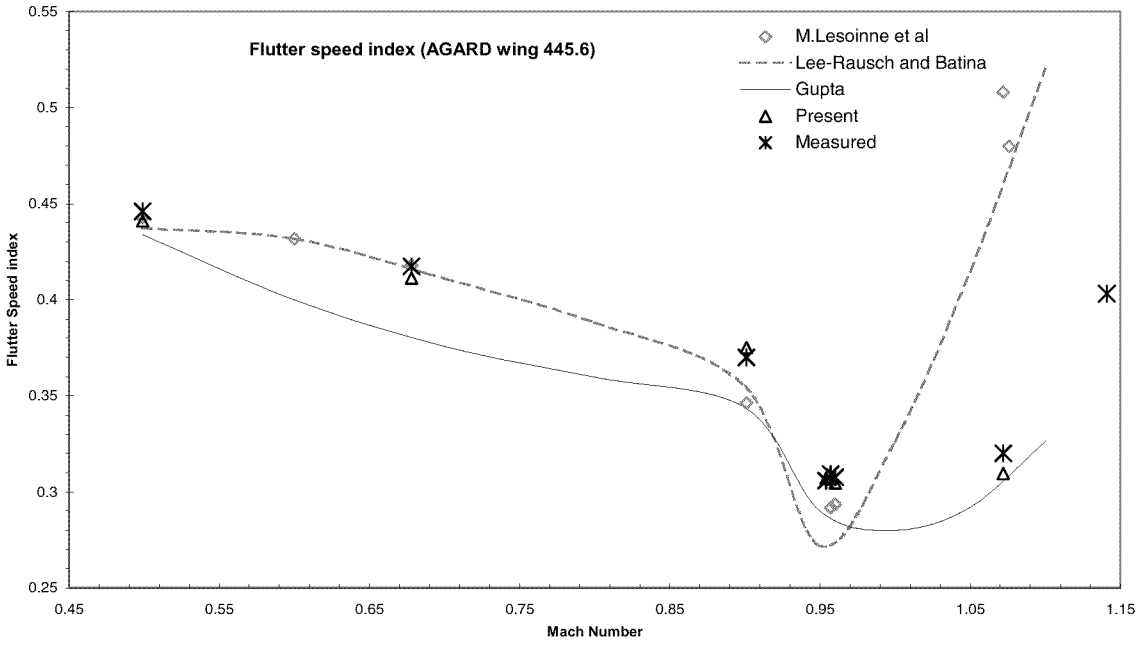

Figure 8. Flutter speed index (AGARD wing 445.6)

Paper #45
Discussor's Name: M. Khalid
Author's Name: A. Soulaimani

Q: When you transfer the aerodynamic loads onto the structural mesh, do you implement any transformation criteria which takes into account the local shear moment and torque equilibrium?

A: Such loads are due to pressure and viscous shear on the skin. In case of a locally attached mass (engine or store), it can be mechanically equivalent to a local force and a torque.